Common Criteria Protection Profile
Package Safe Loader

Kristian Beilke Volker Roth

Arbeitsgruppe Sichere Identität
Freie Universität Berlin
Takustr. 9, 14195 Berlin, Germany
{kristian.beilke, volker.roth}@fu-berlin.de

July 23, 2014

Contents

1 Package Safe Loader

1.1 PP Introduction

This package describes additional security requirements used for in-the-field Software Updates. The package may augment a Protection Profile (hereafter *"original PP"*) with the additional functionality to load software. The original PP describes a combination of a Security Integrated Circuit (Security IC) and a Card Operating System (COS) or Embedded Software, usually in the form of a composite TOE. The original PP is designed with a specific Application or use case in mind and defines a Security IC product. Such a product can be a smart card, an USB token or other devices. If the use case includes the ability to load software or is compatible with the functionality of such a loader, the ST writer shall use this package augmenting the TOE to provide in-the-field Software Updates. **1**

1.1.1 PP Reference

Title:	Protection Profile Package "In-the-filed Software Updates"
Editor(s):	AG-SI FU Berlin
CC Version:	3.1 (Revision 4)
Assurance Level:	Assurance Level for this Protection Profile is EAL4
General Status:	Draft
Version Number:	0.2 as of 06. May 2014
Keywords:	card operating system, Security IC, loader, in-the-field updates

2

1.2 TOE Overview

1.2.1 TOE definition and operational usage

In addition to the TOE definition given in the original Protection Profile the TOE does provide functionality to download data into the FLASH or EEPROM memory. This functionality will be called loader in the following. The loader use extends the usual patch process of the TOE in life cycle phases up to 6 as it is active after delivery of the TOE in its operational usage (in Phase 7: Operational Usage of the TOE life cycle). It enables updates in an environment not necessarily controlled by a trusted party but by the end user or as the case may be even by an attacker. The update mechanism therefore requires special protection measures and safety properties such that a continued safe operational state can be guaranteed. The same loader may also be used in earlier life cycle phases of the TOE, if applicable. **3**

The loader affects data and code stored in non-volatile memory. This includes parts of the firmware (IC-Dedicated Software), the COS (Embedded Software), and the applications which together make up the software of the TOE in the original PP. **4**

All code and data downloaded by the loader is treated as User data before **5**

it is applied or installed. During the installation it is determined if the download is code and/or data and to which part of the TOE it is applied.

The TOE does include data to protect the download functionality. This package does not mandate how this data is stored technically but considers requirements how access to this data is limited. **6**

This package does not dictate a precise technical implementation of the required update mechanism, as this will depend on the available hardware and the security requirements of the applications available on the TOE. It does mandate properties of the mechanism that must be fulfilled. **7**

1.2.2 TOE major security features and operational use

In addition to the functionality specified in BSI-CC-PP-0084 [2] a Security IC product including this package provides the following security functionality: **8**

1. the loader - a mechanism to update Security IC software and/or firmware on in-the-field deployed Security IC products
2. access to the update functionality in form of the loader is only granted to the role of an authenticated update provider, that establishes a trusted channel over which updates are provided
3. the updates are protected for confidentiality and integrity
4. a failed or interrupted update process does not interfere with the operability of the Security IC product, it is always in a working condition according to its life cycle
5. the role of the authenticated update provider can determine the full update version history of the software on the TOE
6. the role of the authenticated update provider can determine the current version of the software.

Application note 1: The first three points of functionality are also covered by BSI-CC-PP-0084 [2] in the Package "Loader dedicated for usage by authorized users only". However [2] does not specify any functional requirements on the loader properties, which motivates the definition of this package. **9**

Application note 2: The functionality to determine the full update history and the current software version can technically be realized as the same requirement. In cases where the software is strictly controlled by a single entity and only a straight update path is allowed the current version directly contains the information which updates have been integrated. In more complex scenarios with multiple entities providing different updates to different parts of the software a more elaborate scheme to determine the integrated updates has to be provided. **10**

Application note 3: This package ignores the technical aspects of applying the update into non volatile memory. The ST writer will need to specify additional details of how software will be integrated. This is especially relevant for **11**

encrypted storage as in certain cases code might need to be encrypted for each single product requiring unique identification.

Application note 4: This package does not contain a security functional requirement to limit the loaders availability and capability. Limited availability can be equated to an explicit disablement of the loader. Limited functionality should guarantee that the loader can not be misused to compromise sensitive data directly instead of indirectly through the loading of malicious code. Both, the capability and the availability, can be affected through an update as long as the loader functionality is not completely implemented in write-protected memory such as the ROM. In such a case the ST writer may include such limiting security requirements. **12**

1.2.3 TOE type

The TOE type is a Security IC product with an update functionality (called loader) which is outside the typical functionality of the Embedded Software or Card Operating System (COS). All further COS functionality is outside the scope of this Package. The original PP defines the TOE as a composite evaluated product which includes the Security IC (base PP) and the Embedded Software for a certain application. **13**

1.2.4 Required non-TOE hardware/software/firmware

For updates to be provided an update service is needed. Access to the update service has to be supplied by the operational environment. The service has access to a terminal which is physically able to communicate with the TOE. In this way the update process can be executed between the service and the loader part of the TOE. **14**

1.3 Security Problem Definition

1.3.1 Assets and External Entities

The assets for this package will be similar to the assets defined in the original PP. In general these will be comprised of protected information that is secret or integrity sensitive such as cryptographic key material and User data but also data representing access rights. In the context of this package a downloaded update is seen as User data while it is stored before it is applied (installed) and becomes part of the TSF or regular user data. In the stored state the update is part of the assets to protect. The assets for this package are listed in Table 1. **15**

Asset	Definition
Secret data	sensitive data that required to be kept confidential
Integrity protected data	data and code required to be kept protected from unauthorized modifications
Software update	Code and/or data downloaded from the update provider and temporarily stored on the TOE before being applied[1].

Table 1: Data objects to be protected by the TOE as primary assets

This package adds an additional external entity to those defined in the original PP. This entity is the update provider. This service provides the software updates to be applied to the TOE. It has a set of keys available to authenticate to the TOE, establish a trusted communication channel, and provide the updates with verifiable integrity protection and authentication information. This package considers the external entities listed in Table 2. As the update process does not involve a human user as an acting entity, only the distinction between an authenticated device and an unauthenticated user is required. **16**

External entity	Definition
World	Any user independent on identification or successful authentication[2].
Update provider	An external device authenticated by cryptographic operation.

Table 2: External entities [3]

This package adds the following security concerns: **17**

SC1 confidentiality and integrity of the User data (update) in transit (while being loaded from the service onto the TOE).

SC2 confidentiality and integrity of the User Data (update) while stored on the TOE.

SC3 confidentiality of the authentication data to access the update functionality and to verify the integrity of the downloaded data.

SC4 integrity of the TOE software state during and after loading of software in-the-field.

1.3.2 Threats

The update functionality of the TOE added by this package must be secured against attacks that can directly or indirectly lead to the disclosure of the assets. This includes the protection of the authentication data used to execute the update functionality, protection of the update itself, and protection of the update functionality against a malicious use. **18**

The following threats are added by this package: **19**

[1]Before application the update is treated as User data. After the update is successfully applied it may become part of the TOE in the form of TSF or other code. It may also be part of the TSF data or User data.

[2]The user World corresponds to the access condition ALWAYS. An authenticated Device is allowed to use the right assigned for World.

[3]This table defines external entities and subjects. Subjects can be recognized by the TOE. As result of an appropriate identification and authentication process, the TOE creates an 'image' inside and 'works' then with this TOE internal image. From this point of view, the TOE itself perceives only 'subjects' and, for them, does not differ between 'subjects' and 'external entities'. There is no dedicated subject with the role 'attacker' within the current security policy, whereby an attacker might 'capture' any subject role recognised by the TOE.

The TOE shall avert the threat "Abuse of Download Functionality (T.Abuse-Loader)" as specified below. **20**

T.Abuse-Loader **Abuse of Download Functionality**

An Attacker may use the correctly working loader to install a software of his choosing.

An attacker that can download a software of his choosing. Such software can be a malicious application, a legitimate but out of date version of the IC Dependent Software or the COS that contains known security vulnerabilities, or any other software that changes the TOE and affects its security. Alternatively the downloaded software may contain assets known by the attacker, overwriting assets inside the TOE. This threat includes an attacker trying to manipulate the downloaded software while in transit between the update service and the TOE[4].

The TOE shall avert the threat "Integrity of the TOE state (T.Integity-State)" as specified below. **21**

T.Integrity-State **Integrity of the TOE state**

An attacker may try to corrupt the TSF code, TSF data, or user code during the in-the-field download of software to leave the TOE in a non-consistent software state.

A non-consistent state can be the TOE not starting up, only partially starting, having unintended functionality, or even allowing unauthorized access to the assets. It is assumed that the use of the loader transits the TOE from one known and evaluated state onto another. The attacker affects this transition in such a way, that the resulting state differs from the intended evaluated state. These attacks result from the attacker controlling the direct environment of the TOE, including the power supply.

The TOE shall avert the threat "Leaking of Downloaded Software (T.Leaking-Download)" as specified below. **22**

T.Leaking-Download **Leaking of Downloaded Software**

An attacker that may access the software downloaded onto the TOE may gain sensitive insides about the TOE that may help to support an attack on the assets.

[4]This threat is intentionally quite broad. The emphasis is on the authorized use of the update functionality with authentic data. The protection of the communication can be viewed as a separate threat but it is assumed that this is a general threat already considered in the original PP. It is nevertheless mentioned in this package as the required countermeasures against this threat are based on functionality provided by the security functionality from the original PP but used for the purpose of this package.

In-the-field updates require the software to be downloaded from a service onto the TOE before it can be installed. After the download it is stored on the TOE. An attacker gaining access to the stored data may learn information that enables attacks, such as the memory location of a patch which reveals a vulnerability. The attacker may also find out about proprietary functionality which provides specific security measures.

The TOE shall avert the threat "Manipulation of Downloaded Software (T.Manipulation-Download)" as specified below. **23**

T.Manipulation-Download	**Manipulation of Downloaded Software** An attacker that may modify the software downloaded onto the TOE may gain control over some parts of the TOE by influencing the transmitted code and data. This may help to support an attack on the assets.

In-the-field updates require the software to be downloaded from a service onto the TOE before it can be installed. During transmission an attacker may try to modify, insert, delete, or replay all or parts of the data that will be applied by the loader. In this way the attacker may influence the control flow directly through code changes, indirectly through parameter changes, or simply disable parts of the TSF to gain access to the assets.

The TOE shall avert the threat "Wrong update order (T.WrongOrder)" as specified below. **24**

T.WrongOrder	**Wrong update order** Software updates applied out of order can leave the TOE in a non-consistent state. In such a state the protection of the assets can not be guaranteed.

The successful application of a software update is not directly observable by the update service since the TOE does not necessarily confirm a completed operation. Multiple updates which might have dependencies on each other and therefore require a certain order of application may exist. An out-of-order application represent a threat that can lead to an non-consistent state just as an corrupted or manipulated update by an attacker could provoke.

1.3.3 Organizational Security Policies

There are no additional Organizational Security Policies defined for this package. It is assumed, that the original PP includes OSPs for the protection of the TOE during Development and Production. **25**

1.3.4 Assumptions

The update provider must ensure the appropriate "Protection of the Update **26**

Service" while providing updates for in-the-field downloads.

A.Download-Sec **Protection of the Update Service**

It is assumed that the service providing in-the-field updates is secured to protect the software to be applied as well as the authentication data required to download the software onto the TOE.

1.4 Security Objectives

The Security Objectives for the TOE and the Security Objectives for the Operational Environment are supplemented for this package. Therefore the Security Objective Rationale is supplemented as well. **27**

1.4.1 Security Objectives for the TOE

The TOE shall provide "Software Integrity (O.SW-Integrity)" as specified below. **28**

O.SW-Integrity **Software Integrity**

The TOE ensures the integrity of the TSF and the data that is stored in any of its Non-Volatile memories. Any downloaded software is integrated fully or not at all. In case of an incomplete download or installation process the TOE will remain unchanged.

The TOE shall provide "Remote Version Attestation (O.VersionCheck)" as specified below. **29**

O.VersionCheck **Remote Version Attestation**

The TOE must provide the authenticated update provider with version information about its software. The supplied information is akin to a fingerprint, identifying precisely what software in which version is installed and additionally providing history information about the upgrade-path.

The following security objectives are taken from BSI-CC-PP-0082 [3], but modified to apply to the special functionality provided by the package Safe Loader. These objectives are assumed to be part of the original PP, where this package is used. But since their existence can not be ensured they are listed here explicitly. **30**

The TOE shall provide "Confidentiality of internal data (O.Confidentiality)" as specified below. **31**

O.Confidentiality **Confidentiality of internal data**

The TOE must ensure the confidentiality of private keys and other confidential TSF Data especially the authentication data, under the TSF scope of control against attacks with high attack potential.

The TOE shall provide "Integrity of user data (O.Integrity)" as specified below. **32**

O.Integrity **Integrity of user data**

The TOE must ensure the integrity of the downloaded updates to prevent attacks through modified patches.

The TOE shall provide "Authentication of external entities (O.Authentication)" as specified below. **33**

O.Authentication **Authentication of external entities**

The TOE supports the authentication of external devices. The TOE is able to authenticate itself to external entities.

The TOE shall provide "Access Control to the Loader (O.AccessControl)" as specified below. **34**

O.AccessControl **Access Control to the Loader**

The TOE must enforce that only authenticated entities with sufficient access control rights can access the loader. The access control policy of the TOE must bind the access control right of the loader to authenticated entities.

1.4.2 Security Objectives for the Operational Environment

The operational environment of the TOE shall provide "Protection of the Update Service (OE.Download-Sec)" as specified below. **35**

OE.Download-Sec

Security procedures shall be used at the Update Service in order to maintain confidentiality and integrity of the TOE updates and the authentication data.

1.4.3 Security Objectives Rationale

A detailed justification required for suitability of the security objectives to coup with the security problem definition is given below. **36**

	O.Confidentiality	O.Integrity	O.Authentication	O.AccessControl	O.SW-Integrity	O.VersionCheck	OE.Download-Sec
T.Leaking-Download	X						
T.Manipulation-Download		X					
T.Abuse-Loader			X	X			
T.Integrity-State					X		
T.WrongOrder						X	
A.Download-Sec							X

Table 3: Security Objective Rationale for the package Safe Loader

The threat **T.Leaking-Download** addresses the compromise of sensitive TSF code and data of an update by an attacker. This threat is prevented by **O.Confidentiality** that ensures the confidentiality of user data, the security services and the TSF data stored on the TOE. **37**

The threat **T.Manipulation-Download** addresses the manipulation of the downloaded user data in transit by an attacker. This is prevented by **O.Integrity** that ensures the integrity of user data. **38**

The threat **T.Abuse-Loader** addresses modification of executable code and TSF data by an attacker through the correct use of the loader with code and data of his choosing. The security objective **O.Authentication** enforces user authentication. In connection with **O.AccessControl** this prevents the unauthorized use of the loader by enforcing access control on authorized users. **39**

The threat **T.Integrity-State** addresses modification of executable code and TSF data by an attacker that manipulates the TOEs software state. This is prevented by **O.SW-Integrity** that ensures the integrity of code and data as it only allows proper transitions between software states before and after an update. **40**

The threat **T.WrongOrder** addresses the application of multiple updates in a wrong order which might lead to a corruption of the TOE's software state. This is prevented by **O.VersionCheck** that ensures the the update service can always determine the version of the software before sending an update to be applied. **41**

The assumption **A.Download-Sec** assumes that the service providing the in-the-field updates is secured. The service protects the software to download onto the TOE and the authentication data required to access the TOE. This assumption is directly addressed by the security objective for the operational environment **OE.Download-Sec**. **42**

1.5 Extended Components Definition

This protection profile uses components defined as extensions to Common Criteria part 2 [1]. The families FIA_API and FPT_ITE are defined in the document on hand. These definitions are taken over from BSI-CC-PP-0082 [3]. **43**

1.5.1 Definition of the Family FIA_API

To describe the IT security functional requirements of the TOE a sensitive family (FIA_API) of the Class FIA (Identification and authentication) is defined here. This family describes the functional requirements for the proof of the claimed identity for the authentication verification by an external entity where the other families of the class FIA address the verification of the identity of an external entity. **44**

Application note 5: The other families of the Class FIA describe only the authentication verification of users' identity performed by the TOE and do not describe the functionality of the user to prove their identity. The following paragraph defines the extended family FIA_API from point of view of a TOE proving its identity. **45**

FIA_API Authentication Proof of Identy **46**

Family Behaviour

This family defines functions provided by the TOE to prove its identity and to be verified by an external entity in the TOE IT environment.

Component leveling:

FIA_API.1 Authentication Proof of Identity, provides prove of the identity of the TOE to an external entity:

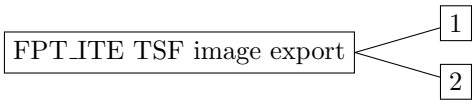
Management:	The following actions could be considered for the management functions in FMT: Management of authentication information used to prove the claimed identity
Audit:	There are no actions defined to be auditable
FIA_API.1	**Authentication Proof of Identity**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_API.1	The TSF shall provide a [assignment: *authentication mechanism*] to prove the identity of the [assignment: *authorized user or role*].

1.5.2 Definition of the Family FPT_ITE TSF image export

Family Behaviour

The family FPT_ITE (TSF image export) of the class FPT (Protection of the TSF) is defined here to describe the IT security functional requirements of the TOE. This family defines rules for fingerprints of TOE implementation and export of TSF data in order to allow verification of their correct implementation in the TOE. The export of a fingerprint of the TOE implementation, e.g. a keyed hash value over all implemented executable code, provides the ability to compare the implemented executable code with the known intended executable code. The export of all non-confidential TSF data, e.g. data security attributes of subjects and objects and public authentication verification data like public keys, provides the ability to verify their correctness e.g. against a specification. The exported TSF images must be correct, but do not need protection of confidentiality or integrity if the export is performed in a protected environment. This family describes the functional requirements for unprotected export of TSF data and export of TOE implementation images not being addressed by any other component of CC part 2 [1]. **47**

Component leveling:



FPT_ITE.1 Export of TOE implementation fingerprint, provides the ability to export the TOE implementation fingerprint without protection of confidentiality or integrity. **48**

FPT_ITE.2 Export of TSF data, provides the ability to export the TSF data without protection of confidentiality or integrity. **49**

Management:	There are no management activities foreseen.
Audit:	There are no actions defined to be auditable

FPT_ITE.1	**Export of TOE implementation fingerprint**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_ITE.1.1	The TOE shall export fingerprint of TOE implementation given the following conditions [assignment: *conditions for export*].
FPT_ITE.1.2	The TSF shall use [assignment: *list of generation rules to be applied*] for the exported data.

FPT_ITE.2	**Export of TSF data**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_ITE.2.1	The TOE shall export [assignment: *list of types of TSF data*] given the following conditions [assignment: *conditions for export*].
FPT_ITE.2.2	The TSF shall use [assignment: *list of encoding rules to be applied by TSF*] for the exported data.

1.6 Security Requirements for Package Safe Loader

1.6.1 Overview

In order to give an overview of the security functional requirements in the context of the security services offered by the TOE, the author of the PP defined the security functional groups and allocated the functional requirements described in the following sections to them. Table 4 list the groups and the corresponding SFRs for this PP. This is not a mapping of SFRs to security objectives as given in section 1.7 but rather a logical organization. **50**

Security Functional Groups	Security Functional Requirements concerned
General Protection of User data and TSF data (section 1.6.3)	FDR_RIP.2, FPT_FLS.1, FPT_TDC.1, FPT_ITE.1, FPT_ITE.2, FPT_TST.1
Authentication (section 1.6.4)	FIA_ATD.1, FIA_UAU.1, FIA_UAU.4, FIA_UAU.5, FIA_UAU.6, FIA_UID.1, FIA_API.1, FMT_SMR.1, FIA_USB.1
Access Control (section 1.6.5)	FDP_ACC.2, FDP_ACF.1, FMT_MSA.3, FMT_SMF.1, FMT_MSA.1, FMT_MTD.1
Protection of communication (section 1.6.6)	FTP_ITC.1, FDP_UCT.1, FDP_UIT.1
Software Integrity Protection (section 1.6.7)	FPT_RCV.4

Table 4: Security functional groups vs. SFRs

Data	Definition
TSF Data	Application Data required to provide update functionality, Version history, Authentication reference data, and Initialization Data integrated to the TOE in the life-cycle phases before End-Usage.
User Data	Downloaded updates, that are not yet applied and integrated into the TSF and TSF Data.

Table 5: TSF Data, User Data for package Safe Loader

1.6.2 Users, subjects and operations

This package does support at least one authentication mechanism based on one set of authentication reference data. A device may be associated with a symmetric cryptographic authentication key and therefore the role gained by the subject acting for this device after successful authentication. The role is defined by the access control rules of the update service. A device may also be associated with a certificate containing the public key as authentication reference data and the card holder authorization (CHA). The authentication protocol comprise the verification of the certificate by means of the root public key and by means of the public key contained in the successful verified certificate. The subject acting for this device gets the role of the CHA which is referenced in the access control rules of the service. Since this package needs to handle only the update service, there is no explicit need for different roles. A successful authenticated entity has the right to use the loader. Unauthenticated entities have no access rights. **51**

Table 6 lists the operations supported by the TOE and the Authentication reference data involved for the update service as subject. These operations are not mapped to ISO/IEC 7816 commands to allow multiple different implementations. ST writers will have to specify a precise mapping. Some operations have to be considered optional since their realization depends on the exact technical circumstances of a product and to which location in memory an update is applied. **52**

Authentication reference data	Operations
Symmetric Authentication key	Authentication of update service
Asymmetric Authentication key	Authentication of update service
Trusted channel	Operations for trusted channel
Private Authentication key	Authentication of TOE
Version history data	Version history requests
Symmetric MAC key	Update integrity verification
Asymmetric key for signature verification	Update integrity verification
	Download of update
	Application of update

Table 6: Operations for package Safe Loader

1.6.3 General Protection of User data and TSF data

The TOE shall meet the requirement "Full residual information protection (FDP_RIP.2)" as specified below **53**

FDP_RIP.2	Full residual information protection
Hierarchical to:	FDP_RIP.1 Subset residual information protection.
Dependencies:	No dependencies.
FDP_RIP.2.1	The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: *allocation of the resource to, deallocation of the resource from*] the all objects.

Application note 6: The writer of the Security Target may want to use iterations of FDP_RIP.1 instead of FDP_RIP.2 in order to distinguish between data, which must be deleted already upon deallocation and those which can be deleted upon allocation. It is recommended to delete secret/private cryptographic keys upon deallocation. For secret data deletion upon allocation should be sufficient (depending on the resistance of the concrete TOE against physical attacks). Note that this package allows modifications of software in-the-field during operational use. Therefore it is theoretically possible that a newly created object uses memory areas, which belonged to another object before. Therefore the COS must ensure that contents of the deleted objects are not accessible by reading the new object. **54**

The TOE shall meet the requirement "Failure with preservation of secure state (FPT_FLS.1)" as specified below **55**

FPT_FLS.1	Failure with preservation of secure state
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_FLS.1.1	The TSF shall preserve a secure state when the following types of failures occur: (1) <u>exposure to operating conditions where therefore a malfunction could occur</u> (2) <u>failure detected by TSF according to FPT_TST.1</u>[5] (3) <u>unsuccessful installation of downloaded software patch</u>

The TOE shall meet the requirement "Inter-TSF basic TSF data consistency (FPT_TDC.1)" as specified below. **56**

[5] [assignment: *list of types of failures in the TSF]*

FPT_TDC.1	Inter-TSF basic TSF data consistency
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_TDC.1.1	The TSF shall provide the capability to consistently interpret Card Verifiable Certificate (CVC)[6] when shared between the TSF and another trusted IT product.
FPT_TDC.1.2	The TSF shall use [assignment: *list of interpretation rules to be applied by the TSF*] when interpreting the TSF data from another trusted IT product.

The TOE shall meet the requirement "Export of TOE implementation fingerprint (FPT_ITE.1)" as specified below (Common Criteria Part 2 extended (see section 1.5.2)). **57**

FPT_ITE.1	Export of TOE implementation fingerprint
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_ITE.1.1	The TSF shall export fingerprint of TOE implementation given the following conditions execution of the command to receive the version or update history[7].
FPT_ITE.1.2	The TSF shall use [selection: *list of generation rules to be applied by TSF*] for the exported data.

The TOE shall meet the requirement "Export of TSF data (FPT_ITE.2)" as specified below (Common Criteria Part 2 extended (see section 1.5.2)). **58**

FPT_ITE.2	Export of TSF data
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_ITE.2.1	The TOE shall export (1) all public authentication reference data[8]. given the following conditions (1) no export of secret data, (2) no export of private keys[9].
FPT_ITE.2.2	The TSF shall use [assignment: *list of encoding rules to be applied by TSF*] for the exported data.

The TOE shall meet the requirement "TSF testing (FPT_TST.1)" as specified below. **59**

[6] [assignment: *list of TSF data types]*
[7] [assignment: *conditions for exports]*
[8] [assignment: *list of types of TSF data*]
[9] [assignment: *conditions for export*]

FPT_TST.1	TSF testing
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_TST.1.1	The TSF shall run a suite of selftests during initial start-up[10] to demonstrate the correct operation of the TSF[11].
FPT_TST.1.2	The TSF shall provide authorized users with the capability to verify the integrity of the TSF data[12].
FPT_TST.1.3	The TSF shall provide authorized users with the capability to verify the integrity of the TSF[13].

1.6.4 Authentication

The TOE shall meet the requirement "User attribute definition (FIA_ATD.1)" as specified below. **60**

FIA_ATD.1	User attribute definition
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_ATD.1.1	The TSF shall maintain the following list of security attributes belonging to individual users: (1) for Device: authentication state gained.[14]

The TOE shall meet the requirement "Timing of authentication (FIA_UAU.1)" as specified below. **61**

[10] [selection: *during initial start-up, periodically during normal operation, at the request of the authorized user, at the conditions [assignment: conditions under which self test should occur]*]
[11] [selection: *[assignment: parts of TSF], the TSF*]
[12] [selection: *[assignment: parts of TSF data], TSF data*]
[13] [selection: *[assignment: parts of TSF], TSF*]
[14] [assignment: *list of security attributes*]

FIA_UAU.1	Timing of authentication
Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.
FIA_UAU.1.1	The TSF shall allow: (1) commands with access control rule ALWAYS for the current life cycle status, (2) commands to start authentication, (3) [assignment: *list of additional TSF mediated actions*][15] on behalf of the user to be performed before the user is authenticated.
FIA_UAU.1.2	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application note 7: If the TOE does not define access control limitation for a command than the TOE shall allow the access for anybody (ALWAYS) and the ST author shall list the command in the element FIA_UAU.1.1. **62**

The TOE shall meet the requirement "Single-use authentication mechanisms (FIA_UAU.4)" as specified below. **63**

FIA_UAU.4	Single-use authentication mechanisms
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UAU.4.1	The TSF shall prevent reuse of authentication data related to (1) the update provider authentication with a symmetric or asymmetric key, (2) [assignment: *additional identified authentication mechanisms*][16].

The TOE shall meet the requirement "Multiple authentication mechanisms (FIA_UAU.5)" as specified below. **64**

[15] [assignment: *list of TSF mediated actions*]
[16] [assignment: *identified authentication mechanisms*]

FIA_UAU.5	Multiple authentication mechanisms
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UAU.5.1	The TSF shall provide:

(1) the execution of authentication commands,

(2) a secure messaging channel,

(3) a trusted channel [17]

to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the following rules:

(1) key based authentication mechanisms shall be used for authenticating of the update provider

(2) [assignment: *additional rules describing how the multiple authentication mechanisms provide authentication*][18].

The TOE shall meet the requirement "Re-authenticating (FIA_UAU.6)" as specified below. **65**

FIA_UAU.6	Re-authenticating
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UAU.6.1	The TSF shall re-authenticate the user under the conditions:

(1) each command sent to the TOE after establishing the secure messaging channel by successful authentication shall be verified as being sent by the authenticated update provider,

(2) each message received after establishing the trusted channel by successful authentication shall be verified as being sent by the authenticated upddate provider[19].

Application note 8: The entities establishing a secure messaging channel respective a trusted channel authenticate each other and agree symmetric session keys. The sender of a command authenticates its message by MAC calculation for the command and the receiver of the commands verifies the authentication by MAC verification of commands. The receiver of the commands authenticates its message by MAC calculation and the sender of a command verifies the authentication by MAC verification of responses. If secure messaging is used with encryption the re-authentication includes the encrypted padding in the plaintext **66**

[17] [assignment: *list of multiple authentication mechanisms*]

[18] [assignment: *rules describing how the multiple authentication mechanisms provide authentication*]

[19] [assignment: *list of conditions under which re-authentication is required*]

as authentication attempt of the message sender and verification of the correct padding as authentication verification by the message receiver. The TOE does not execute any command with incorrect message authentication code. The TOE checks each command by secure messaging in encrypt-then-authenticate mode based on a MAC, whether it was sent by the successfully authenticated communication partner. The TOE does not execute any command with incorrect MAC. Therefore, the TOE re-authenticates the communication partner connected, if a secure messaging error occurred, and accepts only those commands received from the initially communication partner.

The TOE shall meet the requirement "Timing of identification (FIA_UID.1)" as specified below. **67**

FIA_UID.1	Timing of identification
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UID.1.1	The TSF shall allow: (1) commands with access control rule ALWAYS for the current life cycle status, (2) [assignment: *list of additional TSF mediated actions*][20] on behalf of the user to be performed before the user is identified.
FIA_UID.1.2	The TSF shall require each user to be successful identified before allowing any other TSF-mediated actions on behalf of that user.

The TOE shall meet the requirement "Authentication Proof of Identity (FIA_API.1)" as specified below (Common Criteria Part 2 extended (see section 1.5.1)). **68**

FIA_API.1	Authentication Proof of Identity
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_API.1.1	The TSF shall provide commands[21] to prove the identity of the TSF itself.[22]

The TOE shall meet the requirement "Security roles (FMT_SMR.1)" as specified below. **69**

[20] [assignment: *list of TSF mediated actions*]
[21] [assignment: *authentication mechanism*]
[22] [assignment: *authorized user or rule*]

FMT_SMR.1	Security roles
Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.
FMT_SMR.1.1	The TSF shall maintain the roles

1. World as unauthenticated user without authentication reference data,
2. Update provider authenticated by means of symmetric key,
3. Update provider authenticated by means of asymmetric key,
4. [assignment: *additional authorized identified roles*][23]

FMT_SMR.1.2	The TSF shall be able to associate users with roles.

The TOE shall meet the requirement "User-subject binding (FIA_USB.1)" as specified below. **70**

[23] [assignment: *the authorized identified roles*]

FIA_USB.1 User-subject binding
Hierarchical to: No other components.
Dependencies: FIA_ATD.1 User attribute definition.

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:

(1) For the device representing the update service the authenticated role authenticated by a CVC or a symmetric key: *authentication state*[24].

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users:

(1) Initial Authentication State is "not authenticated"[25].

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users:

(1) The authentication state is changed to "authenticated" for the specific authentication context when the update provider has successfully authenticated.

(2) All authentication contexts are lost and the authentication state is set to "not authenticated" for all contexts if the TOE is reset.

(3) If an authentication attempt failed the authentication state for the specific context has to be set to "not authenticated".

(4) If failure of secure messaging (not indicated in CLA-byte, or erroneous MAC, or erroneous cryptogram) is detected the authentication status of the update provider in the current context set to "not authenticated"

(5) [assignment: *further rules for the changing of attributes*][26].

1.6.5 Access Control

The TOE shall meet the requirement "Complete access control (FDP_ACC.2)" as specified below. **71**

[24] [assignment: *list of user security attributes*]
[25] [assignment: *rules for the initial association of attributes*]
[26] [assignment: *rules for the changing of attributes*]

FDP_ACC.2	Complete access control
Hierarchical to:	FDP_ACC.1 Subset access control.
Dependencies:	FDP_ACF.1 Security attribute based access control.
FDP_ACC.2.1	The TSF shall enforce the loader SFP[27] on

(1) the user role Update provider

(2) the objects

(a) symmetric key

(b) private asymmetric key

(c) public asymmetric key

(d) ephemeral keys

(e) downloaded software updates

(f) [assignment: *list of further objects*]

and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2 The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

The TOE shall meet the requirement "Security attribute based access control (FDP_ACF.1)" as specified below. **72**

[27] [assignment: *access control SFP*]

FDP_ACF.1 Security attribute based access control
Hierarchical to: No other components.
Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialization.

FDP_ACF.1.1 The TSF shall enforce the loader SFP[28] to objects based on the following:

(1) the subject user Update Provider with the security attribute *authenticated*

(2) the objects

(a) symmetric key
(b) private asymmetric key
(c) public asymmetric key
(d) ephemeral keys
(e) downloaded software update
(f) [assignment: *list of further objects*][29]

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

(1) The update provider is allowed to create the object update to store the downloaded update on the TOE if the security attributes meet the access rules for the operation download of update.

(2) The update provider is allowed to execute the operation to receive the version history according to FPT_ITE.1 if [assignment: *list of security attributes of subjects*].

(3) [assignment: *further list of subjects, objects, and operations among subjects and objects covered by the SFP*][30].

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: None[31].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*].

[28] [assignment: *access control SFP*]
[29] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]
[30] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]
[31] [assignment: *rules, based on security attributes, that explicitly authorize access of subjects to objects]*

The TOE shall meet the requirement "Specification of Management Functions (FMT_SMF.1)" as specified below. **73**

FMT_SMF.1	Specification of Management Functions
Hierarchical to:	No other components.
Dependencies:	No other components.
FMT_SMF.1.1	The TSF shall be capable of performing the following management functions: (1) Management of device authentication reference data, (2) [assignment: *list of further management functions to be provided by the TSF*][32].

The TOE shall meet the requirement "Management of security attributes (FMT_MSA.1)" as specified below. **74**

FMT_MSA.1	Management of security attributes
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1	The TSF shall enforce the loader SFP[33] to restrict the ability to modify[34] the security attributes update functionality lifecycle and authentication reference data[35] to the update provider[36].

The TOE shall meet the requirement "Static attribute initialization (FMT_MSA.3)" as specified below. **75**

FMT_MSA.3	Static attribute initialization
Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.3.1	The TSF shall enforce the loader SFP[37] to provide restrictive[38] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2	The TSF shall allow the none[39] to specify alternative initial values to override the default values when an object or information is created.

The TOE shall meet the requirement "Management of TSF data (FMT_MTD.1)" as specified below. **76**

[32] [assignment: *list of management functions provided by the TSF]*]
[33] assignment: [*access control SFP(s), information flow control SFP(s)*]
[34] [selection:*change_default, query, modify, delete, [assignment: other operations]*]
[35] [assignment: *list of security attributes*]
[36] [assignment: *the authorized identified roles*]
[37] assignment: [*access control SFP(s), information flow control SFP(s)*]
[38] [selection:*restrictive, permissive, [assignment: other property]*]
[39] [assignment: *the authorised identified roles*]

FMT_MTD.1	Management of TSF data
Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles
	FMT_SMF.1 Specification of Management Functions
FMT_MTD.1.1	The TSF shall restrict the ability to

(1) import[40] the

(a) public keys

(b) certificates as authentication reference data[41]

to roles authorized to execute this operation[42]

(2) export TSF data according to FPT_ITE.1[43] the

(a) the software version information[44]

to the update provider[45]

(3) export TSF data according to FPT_ITE.2[46] the

(a) public authentication reference data[47]

to the update provider[48]

(4) export[49] the following TSF data

(a) Private keys

(b) Session keys

(c) Symmetric authentication keys

(d) Private authentication keys

and the following user data

(e) downloaded software update[50]

to nobody[51].

1.6.6 Protection of communication

The TOE shall meet the requirement "Inter-TSF trusted channel **77**

[40] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[41] [assignment: *list of TSF data*]
[42] [assignment: *the authorized identified roles*]
[43] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[44] [assignment: *list of TSF data*]
[45] [assignment: *the authorized identified roles*]
[46] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[47] [assignment: *list of TSF data*]
[48] [assignment: *the authorized identified roles*]
[49] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[50] [assignment: *list of TSF data*]
[51] [assignment: *the authorized identified roles*]

(FTP_ITC.1)" as specified below.

FTP_ITC.1	Inter-TSF trusted channel
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTP_ITC.1.1	The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1.2	The TSF shall permit another trusted IT product[52] to initiate communication via the trusted channel.
FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for all loader functionality[53].

Application note 9: The TOE responds initially only to commands establishing secure messaging channels. After such a channel is established it is used for the commands to the loader to download and apply updates through the channel. **78**

The TOE shall meet the requirement "Basic data exchange confidentiality (FDP_UCT.1)" as specified below. **79**

FDP_UCT.1	Basic data exchange confidentiality
Hierarchical to:	No other components.
Dependencies:	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] [FDP_ACC.1 Subset access control, or FDP_IFF.1 Subset information flow control]
FDP_UCT.1.1	The TSF shall enforce the loader SFP[54] to receive[55] user data in a manner protected from unauthorized disclosure.

The TOE shall meet the requirement "Data exchange integrity (FDP_UIT.1)" as specified below. **80**

FDP_UIT.1	Data exchange integrity
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFF.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path]
FDP_UIT.1.1	The TSF shall enforce the loader SFP[56] to receive[57] user data in a manner protected from modification, deletion, insertion[58] errors.
FDP_UIT.1.2	The TSF shall be able to determine on receipt of user data, whether modification, deletion, insertion[59] has occurred.

[52] [selection: *the TSF, another trusted IT product*]
[53] [assignment: *list of functions for which a trusted channel is required*]
[54] [assignment: *access control SFP(s) and/or information flow control SFP(s)]*
[55] [selection: *transmit, receive*]
[56] [assignment: *access control SFP(s) and/or information flow control SFP(s)]*
[57] [selection: *transmit, receive*]
[58] [selection: *modification, deletion, insertion, replay*]
[59] [selection: *modification, deletion, insertion, replay*]

1.6.7 TSF Integrity Protection

The TOE shall meet the requirement "Function recovery (FPT_RCV.4)" as specified below. **81**

FPT_RCV.4	Function recovery
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_RCV.4.1	The TSF shall ensure that the function to install a software update[60] have the property that the function either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.

1.7 Security Requirements Rationale

	O.Confidentiality	O.Integrity	O.Authentication	O.AccessControl	O.SW-Integrity	O.VersionCheck
FDR_RIP.2	X					
FPT_FLS.1	X				X	
FPT_TDC.1			X			
FPT_ITE.1	X					X
FPT_ITE.2	X					
FPT_TST.1	X					
FIA_ATD.1			X			
FIA_UAU.1			X			
FIA_UAU.4			X			
FIA_UAU.5			X			
FIA_UAU.6			X			
FIA_UID.1			X			
FIA_API.1			X			X
FMT_SMR.1			X	X		
FIA_USB.1			X	X		
FDP_ACC.2				X		
FDP_ACF.1				X		
FTP_ITC.1	X	X				
FDP_UCT.1	X					
FDP_UIT.1		X				
FPT_RCV.4					X	

Table 7: Mapping between security objectives for the package and SFR

A detailed justification required for *suitability* of the security functional requirements to achieve the security objectives is given below and illustrated in **82**

[60] [assignment: *list of functions and failure scenarios*]

Table 7.

The security objective **O.Confidentiality** "Confidentiality of internal data" requires the protection of the confidentiality of sensitive user data and TSF data. This objective is addressed by the SFRs FDP_RIP.2, FPT_FLS.1, FPT_TST.1, and FDP_UCT.1. FPT_ITE.1 and FPT_ITE.2 restrict the ability to export sensitive TSF data to dedicated roles, some sensitive user data such as private and secret keys are not allowed to be exported at all. Further, FDP_RIP.2 requires that residual information regarding sensitive data in previously used resources will not be available after its usage. FPT_TST.1 requires self tests to demonstrate the correct operation of the TSF and its confidentiality protection capabilities. In case of failures, FPT_FLS.1 requires the preservation of a secure state in order to protect the user data, TSF data and security services. FDP_UCT.1 requires user data in transport to be protected against disclosure. FTP_ITC.1 requires that the TSF provides a communication channel between itself and another trusted IT product. The channel provides assured identification of its end points and protection of the channel data against modification and disclosure. **83**

The security objective **O.Integrity** "Integrity of user data" requires protection against changes to the user data transmitted to the TOE. This objective is addressed by FDP_UIT.1 and FTP_ITC.1. FDP_UIT.1 requires the TSF to follow a SFP for receiving user data. FTP_ITC.1 requires that the TSF provides a communication channel between itself and another trusted IT product. The channel provides assured identification of its end points and protection of the channel data against modification and disclosure. **84**

The security objective **O.Authentication** "Authentication of external entities" requires the support of authentication of external devices as well as the ability of the TSF to authenticate itself. This objective is addressed by the following SFRs: **85**

- FIA_ATD.1 requires that the TSF maintains dedicated security attributes belonging to individual users.
- FIA_UAU.1 requires the processing of dedicated actions before a user is authenticated. Any other actions shall require user authentication.
- FIA_UAU.4 requires the prevention of reuse of authentication data.
- FIA_UAU.5 requires the TSF to support user authentication by providing dedicated commands. Multiple authentication mechanisms like symmetric and asymmetric key based authentication are required.
- FIA_UAU.6 requires the TSF to support re-authentication of message senders using a secure messaging channel.
- FIA_UID.1 requires the processing of dedicated actions before a user is identified. Any other actions shall require user identification.
- FIA_API.1 requires that the TSF provides dedicated commands to prove the identity of the TSF itself.

- FMT_SMR.1 requires that the TSF maintains roles and associates users with roles.
- FIA_USB.1 requires that the TSF associates dedicated security attributes with subjects acting on behalf of that user. Also, the TSF shall enforce rules governing changes of these security attributes by the implementation of commands that perform these changes.

The security objective **O.AccessControl** "Access Control to the Loader" requires the enforcement of an access control policy to restricted objects and devices. Further, the management functionality for the access policy is required. This objective is addressed by the following SFRs: **86**

- FMT_SMR.1 requires that the TSF maintains roles and associates users with roles.
- FIA_USB.1 requires that the TSF associates dedicated security attributes with subjects acting on behalf of that user. Also, the TSF shall enforce rules governing changes of these security attributes by the implementation of commands that perform these changes.
- FDP_ACC.2 requires that the TSF enforces an access control policy to restrict operations on keys performed by subjects of the TOE.
- FDP_ACF.1 requires that the TSF enforce an access control policy to restrict operations on keys based on a set of rules defined in the SFR.

The security objective **O.SW-Integrity** "Software Integrity" requires the ability of the TSF to enforce a safe transition between the TOE software before a downloaded update is installed and after. This objective is addressed by the following SFRs: **87**

- FPT_FLS.1 requires the preservation of a secure state in order to protect the user data, TSF data and security services.
- FPT_RCV.4 requires that the TSF function that installs an update either completes successfully or will recover to a consistent and secure state.

The security objective **O.VersionCheck** "Remote Version Attestation" requires the ability of the TSF to provide an authenticated subject on demand with an unambiguous version information about the TSF software. This either must identify a single version or comprises a history of all applied updates. This objective is addressed by the following SFRs: **88**

- FIA_API.1 requires that the TSF provides dedicated commands to prove the identity of the TSF itself.
- FPT_ITE.1 requires that the TOE exports a fingerprint of the current TOE implementation given corresponding command.

1.7.1 Rationale for SFR's Dependencies

The dependency analysis for the security functional requirements shows that the basis for mutual support and internal consistency between all defined functional requirements is satisfied. All dependencies between the chosen functional components are analyzed, and non-dissolved dependencies are appropriately explained. **89**

The dependency analysis has directly been made within the description of each SFR in section 1.6 above. All dependencies being expected by CC part 2 and by extended components definition in section 1.5 are either fulfilled or their non-fulfillment is justified. **90**

Table 8 lists the required dependencies of the SFRs of this package and gives the concrete SFRs from this document which fulfill the required dependencies. **91**

SFR	dependent on	fulfilled by
FDP_RIP.2	No dependencies.	n.a.
FDP_FLS.1	No dependencies.	n.a.
FDP_TDC.1	No dependencies.	n.a.
FDP_ITE.1	No dependencies.	n.a.
FDP_ITE.2	No dependencies.	n.a.
FDP_TST.1	No dependencies.	n.a.
FIA_ATD.1	No dependencies.	n.a.
FIA_UAU.1	FIA_UID.1 Timing of identification.	FIA_UID.1
FIA_UAU.4	No dependencies.	n.a.
FIA_UAU.5	No dependencies.	n.a.
FIA_UAU.6	No dependencies.	n.a.
FIA_UID.1	No dependencies.	n.a.
FIA_API.1	No dependencies.	n.a.
FMT_SMR.1	FIA_UID.1 Timing of identification.	FIA_UID.1
FIA_USB.1	FIA_ATD.1 User attribute definition.	FIA_ATD.1
FDP_ACC.2	FDP_ACF.1 Security attribute based access control.	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialization.	FDP_ACC.2, FMT_MSA.3
FMT_MSA.3	FMT_MSA.1 Management of security attributes, FMT_SMR.1 Security roles.	FMT_MSA.1, FMT_SMR.1
FDP_SMF.1	No dependencies.	n.a.
FMT_MSA.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_SMR1 Security roles, FMT_SMF.1 Specification of Management Functions.	FDP_ACC.2, FMT_SMR.1, FTM_SMF.1
FMT_MTD.1	FMT_SMR1 Security roles, FMT_SMF.1 Specification of Management Functions.	FMT_SMR.1, FTM_SMF.1

SFR	dependent on	fulfilled by
FTP_ITC.1	No dependencies.	n.a.
FDP_UCT.1	[FDP_ACC.1 Subset access control, or FDP_IFF.1 Subset information flow control], [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path].	FDP_ACC.1, FTP_ITC.1
FDP_UIT.1	[FDP_ACC.1 Subset access control, or FDP_IFF.1 Subset information flow control], [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path].	FDP_ACC.1, FTP_ITC.1
FPT_RCV.4	No dependencies.	n.a.

Table 8: Dependencies of the SFRs

References

[1] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; CCMB-2012-09-002, Version 3.1, Revision 4, September 2012.

[2] Eurosmart. Security Integrated Circuit Platform Protection Profile with Augmentation Packages developed by Inside Secure, Infineon Technologies AG, NXP Semiconductors, STMicroelectronics, Registered and Certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under Reference BSI-CC-PP-0084-2014, Version 1.0, 19.02.2014, February 2014.

[3] T-Systems GEI GmbH. Common Criteria Protection Profile Card Operating System Generation 2 (PP COS G2), Approved by the Federal Office of Information Security (BSI) under Reference BSI-CC-PP-0082, Version 1.0, 23.08.2013, August 2013.